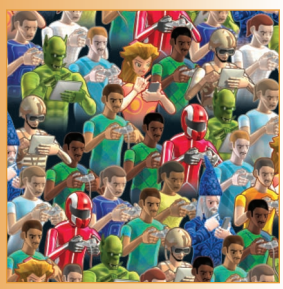


Clouds + Games: A Multifaceted Approach

The computer games landscape is changing: people play games on multiple computing devices with heterogeneous form-factors, capability, and connectivity. Providing high playability on such devices concurrently is difficult. To enhance the gaming experience, designers could leverage abundant and elastic cloud resources, but current cloud platforms aren't optimized for highly interactive games. Existing studies focus on streaming-based cloud gaming, which is a special case for the more general cloud game architecture. The authors explain how to integrate techniques from the cloud and game research communities into a complete architecture for enhanced online gaming quality.

Debadatta Mishra
Indian Institute of Technology, Bombay

Magda El Zarki
University of California, Irvine

Aiman Erbad
Qatar University

Cheng-Hsin Hsu
National Tsing Hua University

Nalini Venkatasubramanian
University of California, Irvine

Computer games are tremendously popular, with global revenue exceeding that of the music and publishing industries.[1] We can classify state-of-the-art games in terms of their resource requirements and system layouts: offline/online, number of players, and architectures (see Figure 1). In particular, computer games impose tight requirements on game precision, responsiveness, and fairness, putting pressure on the game system architecture to achieve good playability. Games utilize one of three mainstream architectures: client-server, in which a centralized server manages the game world; peer-to-peer (P2P), in which peers share the management load; and hybrid, in which a centralized server handles sensitive and compute-intensive tasks, while others, such as local screen updates and optimization techniques, are distributed to peers.[2] Clear tradeoffs exist among these three architectures. Client-server architectures suffer from low scalability, inferior fault tolerance, and high cost, whereas P2P architectures suffer from imperfect consistency control, higher cheating potential, and greater implementation complexity. Hybrid architectures partially cope with weaknesses from each approach, but don't entirely solve these drawbacks.

To meet players' increasing expectations for more immersive experiences, we envision a new cloud-based computer game architecture that leverages abundant and inexpensive cloud resources to ensure improved rendering techniques, shorter response times, better precision, and higher fairness. We refer to this new architecture as *cloud gaming*, which enables workload distribution among multiple cloud servers and game clients. Our definition is more general than the popular

 Published by the IEEE Computer Society 1089-7801/14/$31.00 © 2014 IEEE

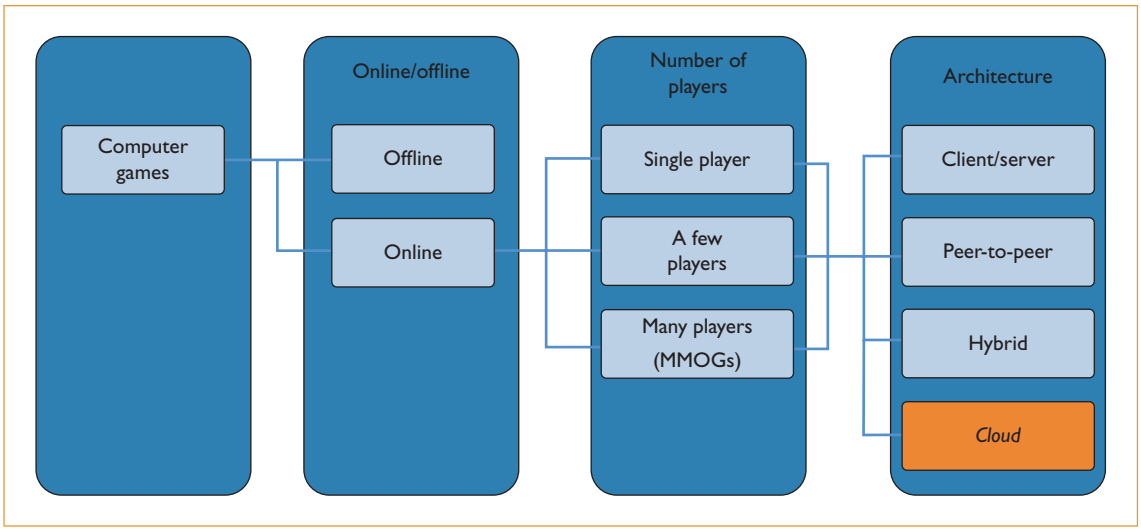


Figure 1. Computer game classification. Various aspects contribute to game classification. The need for game precision, responsiveness, and fairness puts pressure on online games in particular to achieve good playability.

definition in the literature[3] and in the gaming industry (www.sys-con.com/node/2737108), which refers only to streaming-based cloud gaming that moves the whole game engine from a client to a cloud server. A recent study shares the same cloud gaming definition with us, although the authors concentrate on massively multiplayer online games (MMOGs).[4]

By adopting this more general definition, the online game industry could open up a new development realm. First, clouds are scalable in terms of resources, which frees game developers from working under strict resource constraints. Second, clouds are less expensive and offer a more flexible system base for startup companies given the staggering price of purchasing and operating game servers. According to one survey, a company must spend US$0.8 million on servers to support 30,000 concurrent players.[2] Third, clouds are elastic, which is important because games have a very volatile customer base that both varies significantly based on the day/week and can grow rapidly, with some games becoming popular overnight. These unique features make a cloud-based computer game architecture attractive for the game industry.

The landscape for games is also changing in terms of client platforms and networking technologies, especially in mobile settings. Games are no longer confined to powerful machines with high-end wired networking; instead, they're being played on various mobile devices over wireless links. To deal with client diversity, streaming-based cloud game services that assume thin clients have emerged. However, many PC and console games haven't yet migrated to the cloud. The main concern has been latency, because clouds are often remote from players.

Here, we bring the language used in games and that used in clouds into one space to explore how online gaming can be a cloud service while maintaining game quality and playability. We point out several challenges to achieving high-quality games in today's clouds.

Games and Playability

Online games are fairly complex, and it's difficult to get a good sense of game performance when factors such as network latency and device heterogeneity come into play. To develop successful cloud-based games, developers must be better versed in networking concepts and the interplay between network deficiencies and game experience from a user perspective.

Network quality-of-service (QoS) parameters such as delay, jitter, and packet loss are known to influence the user experience or game playability.[5] Quality-of-experience (QoE) metrics constitute a measure of game playability, and their mapping onto network QoS parameters is an ongoing challenge for both online game development and distributed resource provisioning for game deployment. We've developed a novel representation that captures the interplay between

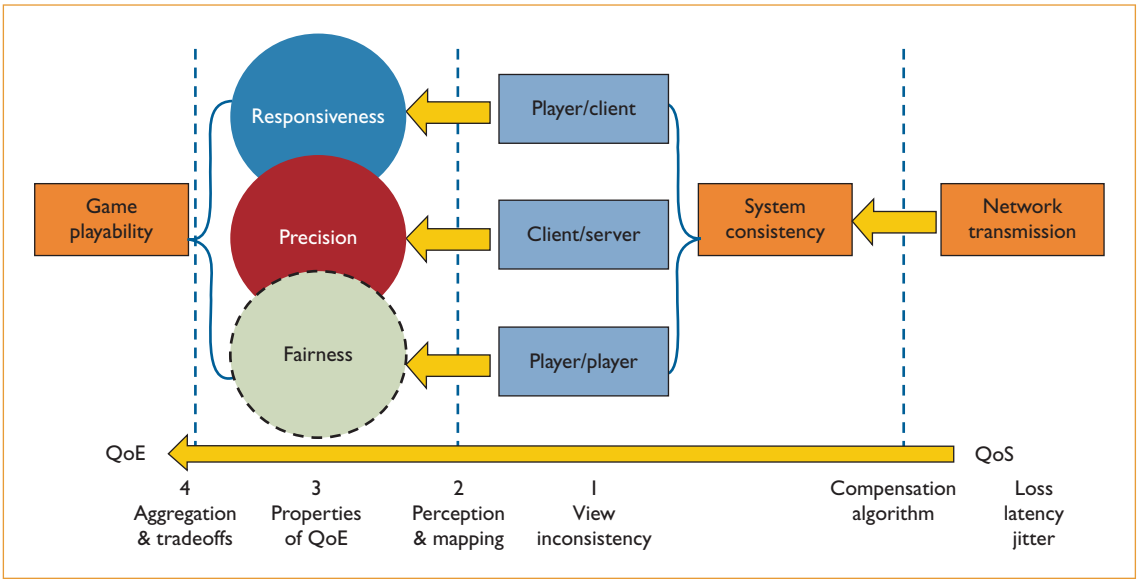


Figure 2. Interplay between quality of service (QoS; right) and quality of experience (QoE; left) metrics. This image bridges the gap between system designs and game playability.

online games' QoS and QoE. Figure 2 shows the network QoS metrics (right) and game QoE metrics (left). *Responsiveness* refers to the time the system takes to respond to a user action.[6] It represents how players perceive the game. We can compute *fairness* as a measure of inconsistency across the game states of different players who are playing in the same game session.[6,7] *Precision* measures the difference between the client and server game states.[5] Games with precise player actions require that this difference be very low for a good player experience.

Responsiveness, fairness, and precision all contribute to a user's perception of a game's QoE. In some games, precision issues can affect user scores by diminishing the number of hits, while low responsiveness can lead to aggravation due to the lag between user actions and when they are displayed on the screen. To deal with some of these issues, game developers have designed optimization techniques that attempt to mitigate the impact of system and network latency on the game experience. To help developers reason about their techniques' effects, we present a game model that captures the main features of an online game and shows the workflow of both client and server game components.

Game Model

We can view a game as one or multiple virtual worlds in which each world is modeled separately. A virtual world is further divided into game zones, which include players and objectives. Examples of objectives can be to shoot an opponent player or pick up a distant health pack. To optimize the user experience, game developers must maximize the number of game actions executed in a given time period. Due to client-side optimization, not every client request needs to go to the server for a response. Thus, the average response time is a valid measure of a game's responsiveness. We can model fairness as the difference between zone states across all clients for a given zone at a given time instance. We derive this from the periodic logs of the zone info. The difference between the client zone state and the server global zone state gives us the measure of precision in the game. More details on the three QoE metrics are available in our prior work.[7]

Figure 3 illustrates online computer games' generic workflow. In each virtual world, a client first checks whether it exists in the zone or has already died. Next, it checks the possibility of predicting the other players' positions to reduce network traffic and improve responsiveness. This flow is part of the client-side optimization techniques that are popular in game development to counter network idiosyncrasies. The player then proceeds with an objective that is enabled given the current location and zone view. The client executes the chosen objective

either locally or remotely based on the type of action required to achieve the objective and on its update frequency with the server.

On the server side of a virtual world, the server receives requests from multiple clients to perform actions. The requests are added to a list of pending requests maintained per server execution window. At the end of each execution window, the server executes requests in the list based on its optimization algorithm, then updates global zone information and sends changes to the clients. These game functions can be offloaded to cloud servers at diverse locations, and many offloading strategies are possible. Choosing the strategy that best trades off network latency, computation power, and network bandwidth is critical to game QoE, which we discuss in detail later.

This game model is a fairly simplified one. Modern games, especially MMOGs, are extremely complex. It isn't directly obvious how we can map the different game functions onto cloud resources while guaranteeing an online game's QoE. Distributing the workflow components among servers in multiple clouds without suffering from high and unpredictable latencies is challenging due to current cloud platforms' limitations, which we look at next.

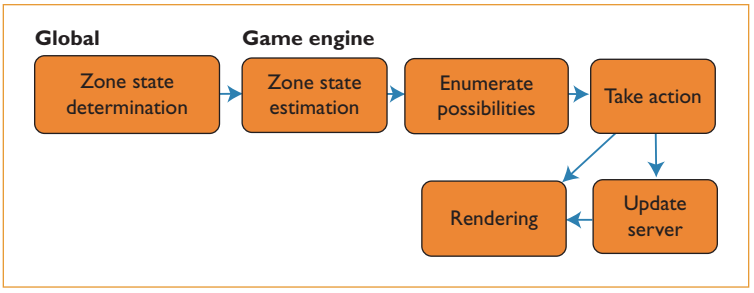


Figure 3. Generic game workflow. When offloading the game functions to cloud servers at diverse locations, it is crucial to choose the offloading strategies to best trade off latency, computation power, and network bandwidth for high game quality of experience (QoE).

Expanding the Current Cloud

The traditional view of cloud platforms focuses primarily on how cloud providers supply resources and services on demand from large resource pools installed in data centers.[8] Such platforms aim to realize economies of scale and increased utilization by sharing resources or services as available through technologies such as virtualization and multitenancy. Examples include Amazon Elastic Compute Cloud (EC2), Google Compute Engine, Windows Azure Cloud Services, and Rackspace. In a multiuser game context, this translates to techniques for offloading compute- and data-intensive tasks from end-clients (who have limited game context and resources) to cloud servers that can gather, assimilate, and process such context. Although public clouds provide resources at scale, a limited number of public cloud data centers are close to users, resulting in large communication latencies in the network infrastructure.

The idea of remotely executing resource-intensive tasks to alleviate resource constraints isn't new. Recent research has focused in particular on mobile cloud applications. Such approaches constantly monitor resource consumption and availability (CPU, network, and so on) to further optimize resource usage. Recent efforts, such as Cloudlets[9] and Mapcloud,[10] have demonstrated the role that local resources close to the user play in ensuring improved application latencies. What's missing in these efforts is an explicit consideration of the QoE that's required to efficiently distribute game workflow components among multiple clouds. In addition, user population fluctuations, if not addressed properly, can result in suboptimal resource mapping choices and ultimately in diminished game QoE.

A new view of the cloud infrastructure is necessary to address the next generation of rich applications such as networked games — a view that supports convergence of the service, compute, communication, and storage infrastructures. This view aims to support a more end-to-end perspective on the information flow from cloud servers (where information is stored and processed) to clients, which use this information for a broad range of applications. In such a view, the networks and their associated servers are key components of the cloud infrastructure. The main idea is to overcome client device and network resource limitations by leveraging available resources in distributed cloud environments. The changing context must also be communicated to users (game clients) accurately and in a timely fashion.

We can thus view the cloud infrastructure as overlay networks that connect end-user devices and multiple data centers. For example, network-as-a-service (NaaS) frameworks[11]

Figure 4. Visual-quality and frame-rate improvements playing the game First-Encounter Assault Recon (FEAR) using cloud servers. (a) The i5 local laptop achieves 7 frames per second (fps) under high rendering quality, which isn't playable. (b) The i5 local laptop can only support low rendering quality. The i7 cloud server always achieves 28+ fps under any rendering quality. These observations demonstrate that cloud gaming can deliver higher frame rates and enable appealing visual effects.

integrate current cloud computing offerings with direct access to the network infrastructure. The idea is to enable tenants or users to easily deploy custom protocols for

- routing,
- multicasting,
- in-transit content editing,
- in-network data aggregation, and
- smart caching.

We must address many open issues to realize this new view of cloud platforms.

Research Challenges

A comprehensive view of the cloud enables group-based collaboration applications, such as cloud games.

Figure 4 illustrates a first-person shooter game (First-Encounter Assault Recon, or FEAR) executing on a streaming-based cloud platform[12] similar to OnLive, where the server runs on an Intel i7 cloud server and the client runs on an Intel i5 laptop. The figure shows that cloud gaming, even in its simplest form, achieves higher frame rates and enables appealing visual effects.

However, we must address multiple research challenges to realize these benefits in a more general and broader setting — for example, over mobile wireless platforms. Techniques are needed to efficiently allocate shared cloud resources across multiple cloud gaming users with common states to achieve high QoE for users and high system utilization for the cloud. Let's examine some of these key problems and shed some light on potential approaches to solving them.

Modeling Games as a Service in the Cloud

Real-time interactive applications such as cloud games are vulnerable to erroneous game states that are due to network or hardware delay and unreliability. Exposing more lower-level system information to cloud games lets developers address quality/cost tradeoffs and prioritize content in games. A couple of questions naturally arise: What is the degree of system/infrastructure awareness required to adequately execute latency-sensitive online games in an outsourced setting? To what extent must online games be aware of the underlying latencies in the network/devices? To answer these questions, we must determine the factors that will influence game behavior and outcome.

To study game behavior, a game designer could vary several parameters at multiple levels:

- Client — number of players, link bandwidth, and execution latency.
- Server — CPU consumption for game logic, server architecture, and access bandwidth.
- Network — topology, access networks, and protocols.
- Game — game actions/objectives and virtual worlds.

On the other hand, multiple performance metrics can capture cloud games' efficacy. Capacity and scalability metrics on the server and network side can help a game designer properly size required resources for game deployment. Game playability metrics determine whether the game will be attractive to users. The designer can then associate the metrics with suitable system parameter values for an online game; cost and deployment constraints dictate how to vary these parameters at different levels to achieve the desired playability metric values. A virtual world, for example, can be modified with fewer bots and reduced details if responsiveness isn't at the expected level.

Unfortunately, the interplay between the allocated resources (CPU and bandwidth utilization,

for example) and their impact on QoE metrics isn't well understood. Given the many alternatives to achieving certain playability targets, the challenge game designers face is choosing the one that best suits the game's purpose and the desired player experience — bearing in mind cost, scalability, and resource constraints. Hence, knowledge of relevant metrics at different levels is required to design optimized state-prediction and resource-allocation strategies.

Provisioning in a Multicloud

In the current cloud market, infrastructure providers have service-level agreements (SLAs) with consumers dictating the resource levels and QoS bounds (in terms of speed, size, bandwidth, and delay); the typical assumption is that devices are interconnected using wired Internet. A primary bottleneck in ensuring QoS with newer platforms and networks (for example, mobile devices connected by last-hop access networks such as 3G and Wi-Fi) is network connectivity's unpredictability. Fundamentally, these networks exhibit varying characteristics. For example, 3G networks offer wide-area ubiquitous connectivity; however, 3G connections can suffer from long delays and slow data transfers, resulting in increased power consumption and cost at the user side. In contrast, Wi-Fi deployments connected to or collocated with Wi-Fi access points — such as 802.11 hotspots — exhibit low communication latencies or delays, and can be used to form a nearby local cloud.[13] Using local-only solutions with Wi-Fi networks, however, creates scalability issues; as the number of users increases, the latency and packet losses increase, causing a decrease in cloud game performance.

One approach would be to synergistically combine local and public cloud capabilities in a two-tier architecture (see Figure 5) to increase cloud games' performance and scalability. In prior work, we successfully implemented such an architecture for mobile applications, called MAPCloud.[10,14] Tier 1 servers in the system architecture represent public cloud services, which are highly scalable and available, but they lack the ability to provide the fine-grained task placement required for low-latency applications such as cloud games. This capability comes from the second-tier local cloud, which consists of servers that are closer (in terms of network distance) to users. For example, each

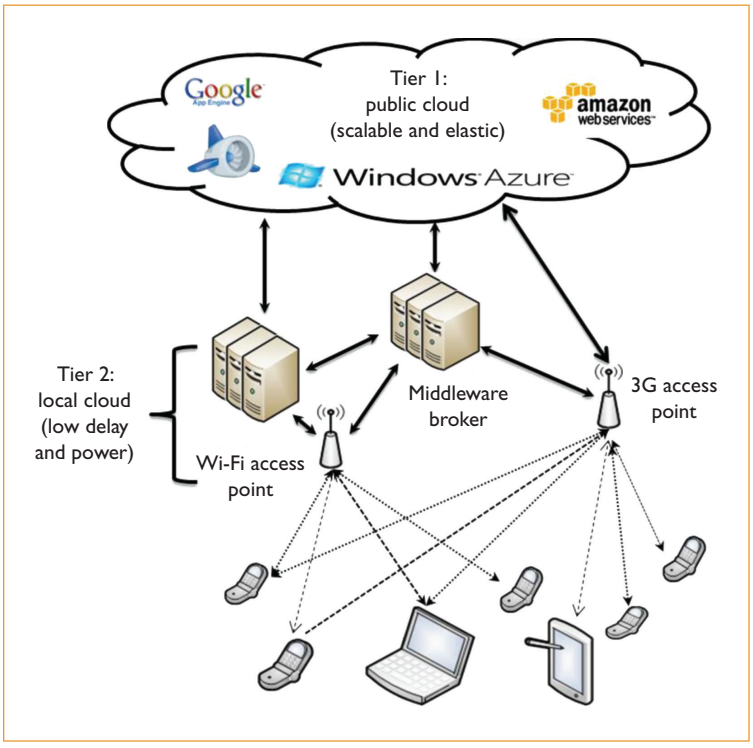


Figure 5. Two-tier multicloud architecture. Tier 1 servers represent public cloud services that are highly scalable and available. Tier 2 consists of servers that are closer to users.

mobile device might be served by a local cloud in close proximity to the connected access point. The main challenge is to intelligently select local and public cloud resources for individual devices to achieve high game playability.

Optimizing resource allocation in the tiered cloud architecture isn't easy. In our past work on mobile cloud computing, we developed a novel framework to model mobile applications as spatiotemporal workflows. Extending our previous work for cloud games is challenging. In general, optimal mapping of workflow tasks to distributed cloud resources is an NP-hard problem. Intuitively, to support latency-sensitive applications, time-sensitive tasks within the workflow must execute closer to where they are needed (for instance, at the client or a local cloud). The challenge is to design efficient algorithms that perform and scale well to a large number of users, while ensuring high game playability.

QoE-Aware Network Adaptation

Large-scale games and rich immersive virtual worlds, such as Second Life, are sensitive to network conditions. Fluctuations at the application level (game demands) and the network level

(available bandwidth in the Internet) result in timing violations. This is especially true when resources are limited. We next discuss potential network adaptation techniques that cloud-game architectures can leverage to cope effectively with limited network bandwidth and wide-area network (WAN) latencies.

At the transport layer, designers can utilize adaptive priority mechanisms to ensure both timeliness and fairness. For example, Paceline is a WAN transport service supporting interactive high-bandwidth multimedia applications.[15] It reduces the latency of high-bandwidth streams in harsh network conditions compared to TCP, and through adaptation keeps the median latency close to the one-way delay for important data. A cloud gaming prototype adopted Paceline to efficiently scale the communications in an epic-scale game scenario,[16] which uses distance as a prioritization criteria.

Adaptive routing of game traffic is also possible at higher network layers. IRS is a detour overlay routing system defined for networked games[17] and motivated by the observation that Internet routing isn't optimized for end-to-end latency. The strategy here is to send game state updates via chosen relay game clients. Experiments with real game traces show promising latency reduction: for example, more than 60 percent of connections enjoy 100+ ms round-trip time reduction. Exploiting resources in public and local clouds can lead to further latency reductions within the cloud gaming architecture. The challenge here is to design suitable algorithms to efficiently locate the best relay servers and clients to maximize game playability.

The issues we've discussed in cloud gaming aren't exhaustive. For example, we didn't address pricing, which remains a tricky problem in the current online gaming setting. With the advent of cloud gaming, the model for pricing becomes even more complex. As the number of users increases, the price per unit of computing would normally decrease due to economies of scale. However, in an environment where the number of users is constantly in flux, determining how to charge is difficult. Usage-based pricing isn't that straightforward in a cloud setting. This and other issues, such as security in the cloud, are among various topics of interest to cloud gaming research.

Cloud computing platforms promise to enhance the gaming experience, but not before we address several research challenges. What's increasingly evident is that the next generation of games and game platforms must execute in dynamic settings; game designers must consider this from the early stages of the design process. The availability of large pools of cloud resources (at a distance) and the simultaneous scarcity of resources (in the local device) are likely to change the landscape of games in the years to come.

References

1. A. Marchand and T. Hennig-Thurau, "Value Creation in the Video Game Industry: Industry Economics, Consumer Benefits, and Research Opportunities," *J. Interactive Marketing*, vol. 27, no. 3, 2013, pp. 141–157.
2. A. Yahyavi and B. Kemme, "Peer-to-Peer Architectures for Massively Multiplayer Online Games: A Survey," *ACM Computing Surveys*, vol. 46, no. 1, 2013, article no. 9.
3. W. Cai, V. Leung, and M. Chen, "Next Generation Mobile Cloud Gaming," *Proc. IEEE 7th Int'l Symp. Service Oriented System Engineering* (SOSE 13), 2013, pp. 551–560.
4. D. Maggiorini and L. Ripamonti, "Cloud Computing to Support the Evolution of Massive Multiplayer Online Games," *Comm. Computer and Information Science*, vol. 220, 2011, pp. 101–110.
5. M. Claypool and K. Claypool, "Latency Can Kill: Precision and Deadline in Online Games," *Proc. 1st Ann. ACM SIGMM Conf. Multimedia Systems* (MMSys 10), 2010, pp. 215–222.
6. J. Brun, F. Safaei, and P. Boustead, "Fairness and Playability in Online Multiplayer Games," *Proc. 3rd IEEE Consumer Comm. and Networking Conf.*, (CCNC 06), 2006, pp. 1199–1203.
7. P. Chen and M. El Zarki, "Perceptual View Inconsistency: An Objective Evaluation Framework for Online Game Quality of Experience (QOE)," *Proc. 10th Ann. Workshop Network and Systems Support for Games* (NetGames 11), 2011, article no. 2.
8. M. Armbrust et al., "A View of Cloud Computing," *Comm. ACM*, vol. 53, no. 4, 2010, pp. 50–58.
9. M. Satyanarayanan et al., "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 1, 2009, pp. 14–23.
10. M. Rahimi, N. Venkatasubramanian, and A. Vasilakos, "MuSIC: On Mobility-Aware Optimal Service Allocation in Mobile Cloud Computing," *Proc. IEEE 6th Int'l Conf. Cloud Computing* (CLOUD 13), 2013, pp. 75–82.

11. P. Costa et al., "NaaS: Network-as-a-Service in the Cloud," *Proc. 2nd Usenix Conf. Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services* (Hot-ICE 12), 2012; www.usenix.org/system/files/conference/hot-ice12/hotice12-final29.pdf.
12. C. Huang et al., "GamingAnywhere: An Open Cloud Gaming System," *Proc. 4th ACM Multimedia Systems Conf.* (MMSys 13), 2013, pp. 36–47.
13. E. Cuervo et al., "MAUI: Making Smartphones Last Longer with Code Offload," *Proc. 8th Int'l Conf. Mobile Systems, Applications, and Services* (MobiSys 10), 2010, pp. 49–62.
14. M. Rahimi et al., "MAPCloud: Mobile Applications on an Elastic and Scalable 2-Tier Cloud Architecture,"*Proc. 2012 IEEE/ACM 5th Int'l Conf. Utility and Cloud Computing* (UCC 12), 2012, pp. 83–90.
15. A. Erbad and C. Krasic, "Sender-Side Buffers and the Case for Multimedia Adaptation," *Comm. ACM*, vol. 55, no. 12, 2012, pp. 50–58.
16. M. Najaran and C. Krasic, "Scaling Online Games with Adaptive Interest Management in the Cloud," *Proc. 9th Ann. Workshop Network and Systems Support for Games* (NetGames 10), 2010, article no. 9.
17. C. Ly, C. Hsu, and M. Hefeeda, "IRS: A Detour Routing System to Improve Quality of Online Games," *IEEE Trans. Multimedia*, vol. 13, no. 4, 2011, pp. 733–747.

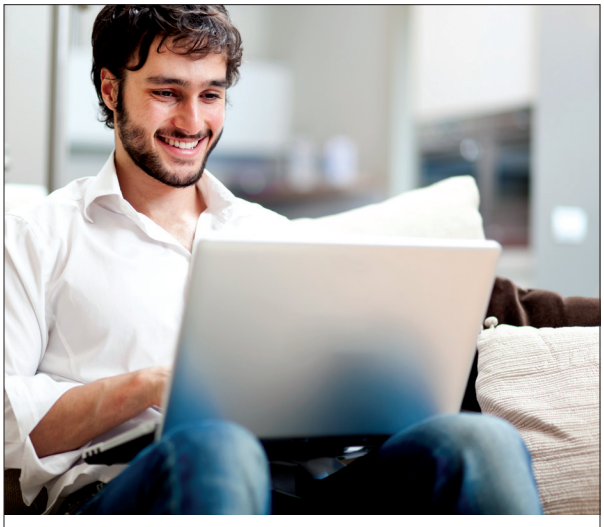
Debadatta Mishra is a PhD student in the Department of Computer Science and Engineering at the Indian Institute of Technology, Bombay. His research interests are in operating systems, virtualization and cloud computing, communication networks, and online network games. Mishra received a masters in computer application from Utkal University, India. Contact him at deba@cse.iitb.ac.in.

Magda El Zarki is a professor in the Department of Computer Science at the University of California, Irvine, and the director of the Computer Games and Virtual Worlds Center. Her research is in telecommunication networks and networked computer games. El Zarki received a PhD in electrical engineering from Columbia University. She is coauthor of the textbook *Mastering Networks – An Internet Lab Manual* (Addison-Wesley, 2004). Contact her at elzarki@uci.edu.

Aiman Erbad is an assistant professor in the Department of Computer Science and Engineering at Qatar University. His research interests include cloud computing, and multimedia systems and networking. Erbad received a PhD in computer science from the University of British Columbia. He's a member of IEEE and ACM. Contact him at aerbad@qu.edu.qa.

Cheng-Hsin Hsu is an assistant professor at National Tsing Hua University, Taiwan. His research interests are in multimedia networking and distributed systems. Hsu received a PhD in computing science from Simon Fraser University, British Columbia. He's a member of IEEE and ACM. Contact him at chsu@cs.nthu.edu.tw.

Nalini Venkatasubramanian is a professor in the School of Information and Computer Science at the University of California, Irvine. Her research interests include distributed and parallel systems, middleware, mobile environments, multimedia systems and applications, and formal reasoning of distributed systems. Venkatasubramanian received a PhD in computer science from the University of Illinois, Urbana-Champaign. She's a member of IEEE and ACM. Contact her at nalini@ics.uci.edu.



Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.